



```
IIIIII  NN      NN  PPPPPPP  UU      UU  TTTTTTTTTT
IIIIII  NN      NN  PPPPPPP  UU      UU  TTTTTTTTTT
  II    NN      NN  PP        PP  UU      UU  TT
  II    NN      NN  PP        PP  UU      UU  TT
  II    NNNN     NN  PP        PP  UU      UU  TT
  II    NNNN     NN  PP        PP  UU      UU  TT
  II    NN  NN   NN  PPPPPPP  UU      UU  TT
  II    NN  NN   NN  PPPPPPP  UU      UU  TT
  II    NN      NNNN  PP       UU      UU  TT
  II    NN      NNNN  PP       UU      UU  TT
  II    NN      NN   PP       UU      UU  TT
  II    NN      NN   PP       UU      UU  TT
IIIIII  NN      NN  PP       UU      UU  TT
IIIIII  NN      NN  PP       UUUUUUUUUU  TT
IIIIII  NN      NN  PP       UUUUUUUUUU  TT
```

```
....
....
....
....
```

```
LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```



(2)	73	DECLARATIONS
(3)	102	MAC\$GETCHR GET NEXT CHARACTER FROM INPUT STREAM
(4)	222	MAC\$GETLIN GET NEXT INPUT SOURCE LINE
(5)	324	OPEN NEXT INPUT SOURCE FILE
(6)	359	STAT4 SWITCH INPUT TO MACRO TEXT
(7)	391	MEXIT MACRO EXIT ROUTINE
(7)	403	MAC\$POP INPUT POP INPUT CONTEXT UP A LEVEL
(8)	447	STATEMENTS END-OF-LINE CLEANUP

B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
W  
X  
Y  
Z  
[  
\  
]  
^  
\_  
`  
a  
b  
c  
d  
e  
f  
g  
h  
i  
j  
k  
l  
m  
n  
o  
p  
q  
r  
s  
t  
u  
v  
w  
x  
y  
z  
{  
|  
}  
~



```
0000 1 .TITLE MAC$INPUT GET NEXT CHARACTER
0000 2 .IDENT 'V04-000'
0000 3
0000 4 :
0000 5 :*****
0000 6 :*
0000 7 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 :* ALL RIGHTS RESERVED.
0000 10 :*
0000 11 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 :* TRANSFERRED.
0000 17 :*
0000 18 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 :* CORPORATION.
0000 21 :*
0000 22 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 :*
0000 25 :*
0000 26 :*****
0000 27 :
0000 28 :
0000 29 :++
0000 30 : FACILITY: VAX MACRO ASSEMBLER OBJECT LIBRARY
0000 31 :
0000 32 : ABSTRACT:
0000 33 :
0000 34 : The VAX-11 MACRO assembler translates MACRO-32 source code into object
0000 35 : modules for input to the VAX-11 LINKER.
0000 36 :
0000 37 : ENVIRONMENT: USER MODE
0000 38 :
0000 39 : AUTHOR: Benn Schreiber, CREATION DATE: 21-AUG-78
0000 40 :
0000 41 : MODIFIED BY:
0000 42 :
0000 43 : V03.01 MTR0011 Mike Rhodes 18-Mar-1982
0000 44 : Add logic to routine MAC$GETLIN to trap non-RMSS$ EOF errors
0000 45 : and non-SUM$ xxx errors. This occurs when an illegal record
0000 46 : is encountered by RMS and it does not count as a SUM error,
0000 47 : which leaves us in an infinite loop trying to read the next line.
0000 48 : Fixes QAR #691.
0000 49 :
0000 50 : V03.00 MTR0006 Mike Rhodes 15-Mar-1982
0000 51 : Modify MAC$GETCHR to allow ALL characters to be passed
0000 52 : when the FLG$V_DLIMSTR flag is set. The characters allowed
0000 53 : to be passed include the semicolon (which was already passed)
0000 54 : and the hyphen, which at times was incorrectly treated as a
0000 55 : line continuation character. Fixes SPR #11-42904.
0000 56 :
0000 57 : V01.10 RN0023 R. Newland 3-Nov-1979
```



0000 58 :  
0000 59 :  
0000 60 :  
0000 61 :  
0000 62 :  
0000 63 :  
0000 64 :  
0000 65 :  
0000 66 :  
0000 67 :  
0000 68 :  
0000 69 :  
0000 70 :  
0000 71 :--

New message codes to get error messages from system  
message file.

V01.09 RN0010 R. Newland 5-Sep-1979  
Multipage MXB blocks

V01.08 RN0005 R. Newland 27-Aug-1979  
Remove .ALIGN LONG statements

V01.07 RN0002 R. Newland 01-Feb-1979  
Changes for Source Update Merge, Get input lines  
from SUM\$LINE.



```

0000 73      .SBTTL DECLARATIONS
0000 74      :
0000 75      : INCLUDE FILES:
0000 76      :
0000 77      :
0000 78      :
0000 79      : MACROS:
0000 80      :
0000 81      $RABDEF      ;DEFINE RAB OFFSETS
0000 82      $MAC_GENVALDEF ;VAX-11 MACRO GENERAL SYMBOLS
0000 83      $MAC_CTLFLGDEF ;CONTROL FLAGS
0000 84      $MAC_INTCODDEF ;INTERMEDIATE CODE FILE SYMBOLS
0000 85      $MAC_INPBLKDEF ;DEFINE INPUT BLOCK OFFSETS
003C 86      $MAC_SYMBLKDEF ;DEFINE SYMBOL BLOCK OFFSETS
0000 87      $MAC_MNBDEF    ;DEFINE MNB OFFSETS
0000 88      $MACMSGDEF     ; Define message codes
0000 89      $STSDEF       ;STATUS BITS
0000 90      DEFSUMCBL     ; Define SUM control block symbols
0000 91      :
0000 92      :
0000 93      :
0000 94      : EQUATED SYMBOLS:
0000 95      :
0000 96      :
0000 97      :
0000 98      : OWN STORAGE:
0000 99      :
0000 100

```



```
0000 102 .SBTTL MAC$GETCHR GET NEXT CHARACTER FROM INPUT STREAM
0000 103 :++
0000 104 : FUNCTIONAL DESCRIPTION:
0000 105 :
0000 106 : THIS ROUTINE IS CALLED WHENEVER A NEW CHARACTER IS NEEDED.
0000 107 : IT PERFORMS ALL THE HOUSEKEEPING FOR ENDS OF LINES, OUTPUTS
0000 108 : COMMANDS TO THE INT. FILE FOR NEW LINES, AND HANDLES CONTINUATION
0000 109 : LINES.
0000 110 :
0000 111 : CALLING SEQUENCE:
0000 112 :
0000 113 : JSB MAC$GETCHR
0000 114 :
0000 115 : INPUT PARAMETERS:
0000 116 :
0000 117 : R10 LAST CHARACTER READ
0000 118 : R11 POINTS TO FLAGS (MAC$GL_FLAGS)
0000 119 : IF FLG$M_ALLCHR IS SET THEN SEMICOLONS WILL BE PASSED
0000 120 : BACK. IF THE FLAG IS CLEAR, SEMICOLONS AND
0000 121 : EVERYTHING FOLLOWING WILL BE IGNORED.
0000 122 :
0000 123 : IMPLICIT INPUTS:
0000 124 :
0000 125 : NONE
0000 126 :
0000 127 : OUTPUT PARAMETERS:
0000 128 :
0000 129 : R10 NEXT CHARACTER
0000 130 :
0000 131 : IMPLICIT OUTPUTS:
0000 132 :
0000 133 : NONE
0000 134 :
0000 135 : COMPLETION CODES:
0000 136 :
0000 137 : NONE
0000 138 :
0000 139 : SIDE EFFECTS:
0000 140 :
0000 141 : NONE
0000 142 :
0000 143 : --
0000 144 :
0000 145 :
0000 146 :
00000000 147 .PSECT MAC$RO_CODE_P15,NOWRT,GBL, LONG
0000 148 :
0000 149 .ENABL LSB
0000 150 :
0000 151 MAC$GETCHR::
58 0000 58 DD 0000 152 PUSHL R8 ;SAVE R8
0D 5A 9E 0002 153 10$: MOVAB W^MAC$GL_LINEPT,R8 ;POINT TO MAC$GL_LINEPT
1F 12 91 0007 154 20$: CMPB R10,#CR ;TIME TO READ NEW LINE?
5A 20 9A 000A 155 30$: BNEQ ;IF NEQ NO
50 0000 5A 000C 156 25$: MOVZBL #BLNK,R10 ;YES--PREVENT LOOPING ON CR
6B 08 C8 000F 157 MOVL W^MAC$GL_INPUTP,R0 ;ADDRESS INPUT BLOCK
0014 158 BISL2 #FLG$M_CONT,(R11) ;ALLOW CONTINUATION LINES
```



```
GET NEXT CHARACTER
MAC$GETCHR GET NEXT CHARACTER FROM INPUT

08 B0 16 0017 159 JSB @INPSL_GETL(R0) ;CALL INPUT ROUTINE TO READ NEXT LINE
E9 6B 10 E0 001A 160 BBS #FLGSV_MACTXT,(R11),20$ ;BRANCH IF READING MACRO TEXT
001E 161 $INTOUT_WD INT$ _NEWL,- ;YES--SIGNAL NEW LINE FOR PASS2
001E 162 <W^MAC$GL_LINENUM>
DC 11 0029 163 BRB 20$ ;CONTINUE
5A 00 B8 9A 002B 164 30$: MOVZBL @ (R8),R10 ;GET NEXT CHARACTER
68 D6 002F 165 INCL (R8) ;POINT TO NEXT CHARACTER
5A D5 0031 166 TSTL R10 ;WAS CHARACTER A NULL?
F6 13 0033 167 BEQL 30$ ;IF EQL YES--GET NEXT CHARACTER
0035 168 ; (GOD ONLY KNOWS WHERE NULLS COME FROM!)
03 6B 2F E1 0035 169 BBC #FLGSV_DLIMSTR,(R11),35$ ;ALLOW SEMICOLONS AND HYPHENS?
007B 31 0039 170 BRW 150$ ;YES -- BYPASS OTHER TESTS.
0E 6B 1A E0 003C 171 35$: BBS #FLGSV_SPLALL,(R11),40$ ;BRANCH IF SHOULD NOT EVEN CONSIDER
0040 172 ;SEMI-COLONS
3B 5A 91 0040 173 CMPB R10,#SEMI ;IS CHARACTER A SEMI-COLON?
09 12 0043 174 BNEQ 40$ ;IF NEQ NO
06 6B E8 0045 175 BLBS (R11),40$ ;YES--AND ARE WE PASSING ALL CHARS.?
0048 176 ;(ALLCHR IS LOW BIT!!)
5A 0D 9A 0048 177 MOVZBL #CR,R10 ;NO--CALL IT EOL
0069 31 004B 178 BRW 150$ ;TAKE THE QUICK EXIT
2D 5A 91 004E 179 40$: CMPB R10,#HYPHEN ;LINE CONTINUATION?
64 12 0051 180 BNEQ 150$ ;IF NEQ NO
60 6B 03 E1 0053 181 BBC #FLGSV_CONT,(R11),150$ ;YES--CONTINUATIONS ALLOWED?
5A DD 0057 182 PUSHL R10 ;YES--SAVE CURRENT STATE
68 DD 0059 183 PUSHL (R8) ;SAVE MAC$GL_LINEPT
7E 6B FFFFFFFE 8F CB 005B 184 BICL3 #^C<FLGSM_ALLCHR>,(R11),-(SP) ;SAVE ALLCHR STATE
6B 09 CA 0063 185 BICL2 #FLGSM_CONT!FLGSM_ALLCHR,(R11) ;DON'T RECURSE ON LINES THAT HAVE
5A 20 9A 0066 186 MOVZBL #BLNK,R10 ; ALL HYPHENS.
20 5A 91 0069 187 60$: CMPB R10,#BLNK ;IS CHARACTER A BLANK?
05 13 006C 188 BEQL 70$ ;IF EQL YES
09 5A 91 006E 189 CMPB R10,#TAB ;NO--IS IT A TAB?
05 12 0071 190 BNEQ 80$ ;IF NEQ NO
FF8A 30 0073 191 70$: BSBW MAC$GETCHR ;YES--SKIP OVER SPACES AND TABS
F1 11 0076 192 BRB 60$ ;FIND NON-BLANK, NON-TAB CHARACTER
0D 5A 91 0078 193 80$: CMPB R10,#CR ;IS THIS EOL?
2E 12 007B 194 BNEQ 130$ ;IF NEQ NO
6B 8E C8 007D 195 BISL2 (SP)+,(R11) ;YES--RESTORE ALLCHR FLAG
5E 08 C0 0080 196 ADDL2 #2*4,SP ;AND SAVED CONTEXT NOT NEEDED
FF7A 30 0083 197 BSBW MAC$GETCHR ;READ NEXT LINE
04 6B 0D E1 0086 198 BBC #FLGSV_OPRND,(R11),90$ ;BRANCH IF NOT IN OPERAND FIELD
06 6B 14 E3 008A 199 BBS #FLGSV_CHKLPND,(R11),100$ ;SET CHKL PENDING AND BRANCH IF NONE PENDI
008E 200 90$: $INTOUT_X INT$ _CHKL ;ENSURE ALIGNMENT OF CONTINUED LINES
0D 5A 91 0094 201 100$: CMPB R10,#CR ;NULL LINE?
0D 13 0097 202 BEQL 120$ ;IF EQL YES
0C 5A 91 0099 203 CMPB R10,#FF ;STILL LOOKING FOR NULL LINES
05 13 009C 204 BEQL 110$
3B 5A 91 009E 205 CMPB R10,#SEMI
11 12 00A1 206 BNEQ 140$
5A 0D 9A 00A3 207 110$: MOVZBL #CR,R10 ;TREAT AS EOL IF NULL
FF57 30 00A6 208 120$: BSBW MAC$GETCHR ;READ NEXT LINE
E9 11 00A9 209 BRB 100$ ;FIND NON-NULL LINE
00AB 210
00AB 211 ; NOT REALLY A CONTINUED LINE
00AB 212
6B 8E C8 00AB 213 130$: BISL2 (SP)+,(R11) ;RESTORE ALLCHR FLAG
68 8ED0 00AE 214 POPL (R8) ;RESTORE MAC$GL_LINEPT
5A 8ED0 00B1 215 POPL R10 ;RESTORE CHARACTER
```



MAC\$INPUT  
V04-000

GET NEXT CHARACTER  
MAC\$GETCHR GET NEXT CHARACTER FROM INPUT

G 16

16-SEP-1984 02:20:18 VAX/VMS Macro V04-00  
5-SEP-1984 01:48:32 [MACRO.SRC]INPUT.MAR;1

Page 6  
(3)

6B	08	C8	00B4	216	140\$:	BISL2	#FLG\$M_CONT,(R11)	;ALLOW CONTINUATIONS AGAIN
	58	8ED0	00B7	217	150\$:	POPL	R8	;RESTORE R8
		05	00BA	218		RSB		;RETURN WITH CHARACTER IN R10
			00BB	219				
			00BB	220		.DSABL	LSB	



```
00BB 222 .SBTTL MAC$GETLIN GET NEXT INPUT SOURCE LINE
00BB 223
00BB 224 :++
00BB 225 THIS ROUTINE IS CALLED TO GET THE NEXT LINE OF THE CURRENT
00BB 226 INPUT FILE AND PLACE IT IN MAC$AB_LINEBF.
00BB 227 :--
00BB 228
00BB 229 .ENABL LSB
00BB 230
00BB 231 MAC$GETLIN::
50 0000'CF 9E 00BB 232 MOVAB W^MAC$INPUT_RAB,R0 ;POINT TO THE RAB
0000'CF DF 00C0 233 PUSHAL W^MAC$GT_SCB ; Control block address
00000000'GF 01 FB 00C4 234 CALLS #1,G^SUM$LINE ; Get next input line
6D 50 E8 00CB 235 BLBS R0,40$ ; If LBS then good read
00000000'8F 50 D1 00CE 236 CMPL R0,#RMS$_EOF ; Was error end-of-file?
27 13 00D5 237 BEQL 8$ ; Yes if EQL, try next file
51 50 0C 10 EF 00D7 238 EXTZV #ST$V_FAC NO,#ST$S_FAC NO,R0,R1 ; Get facility no
0084 8F 51 B1 00DC 239 CMPW R1,#<SUM$_NORMAL-16> ; SUM error?
OC 13 00E1 240 BEQL 2$ ; Yes if EQL
51 50 FFFFFFFF 8F CB 00E3 241 BICL3 #^CST$M_SEVERITY,R0,R1 ; Copy and check the severity
CE 13 00EB 242 BEQL MAC$GETLIN ; WARNING, try again
OF 11 00ED 243 BRB 8$ ; ERROR or FATAL, try next file
05 6B OE E0 00EF 244 2$: BBS #FLG$V_P2,(R11),5$ ; Return line if Pass 2
00AF 30 00F3 245 BSBW SUM_ERROR ; Generate intermediate code if Pass 1
C3 11 00F6 246 BRB MAC$GETLIN
0000'CF D7 00F8 247 5$: DECL W^MAC$GL_LINENUM ; Don't increment line number
3D 11 00FC 248 BRB 40$
00F5 30 00FE 249 8$: BSBW MAC$NXTINPFIL ;ELSE TRY TO OPEN NEXT FILE
0000'CF 01 9A 0101 250 MOVZBL #1,W^MAC$GL_SRC_PAG ;RESET PAGE COUNT TO ONE
B2 50 E8 0106 251 BLBS R0,MAC$GETLIN ;IF THERE IS ANOTHER FILE GO READ IT
0000'CF 444E452E 8F D0 0109 252 10$: MOVL #^A/.END/,W^MAC$AB_LINEBF ;OOPS--NO FILE--FAKE A '.END'
0022'CF 04 9B 0112 253 MOVZBW #4,W^MAC$INPUT_RAB+RAB$W_RSZ ;WITH A SIZE OF 4 BYTES
0000'CF D5 0117 254 TSTL W^MAC$GL_IF_LEVEL ;IN UNFINISHED CONDITIONAL?
OC 15 011B 255 BLEQ 30$ ;IF LEQ NO
0004'CF 43 8F 90 011D 256 MOVB #^A/C/,W^MAC$AB_LINEBF+4 ;YES--MAKE .END INTO .ENDC
0022'CF B6 0123 257 INCW W^MAC$INPUT_RAB+RAB$W_RSZ ;COUNT THE CHARACTER
12 11 0127 258 BRB 40$ ;CONTINUE
OE 6B OE E0 0129 259 BBS #FLG$V_P2,(R11),40$ ;ONLY MESSAGE ON PASS 1
012D 260 $INTOUT_X INT$FNEWL ;PRINT CONTENTS OF PREVIOUS LINE
012D 261 :*** $INTOUT_LW INT$WRN,- ;TELL OF MISSING END STATEMENT
012D 262 <#MAC$_MISSINGEND,#0>
0022'CF 3C 013B 263 MOVZWL W^MAC$INPUT_RAB+RAB$W_RSZ,- ;SAVE LENGTH OF LINE
50 013F 264 R0
0000'CF 50 D0 0140 265 MOVL R0,W^MAC$GL_LINELN ;...
50 0000'CF 9E 0145 266 MOVAB W^MAC$AB_LINEBF,R0 ;GET ADDRESS OF LINE BUFFER
0000'CF 50 D0 014A 267 MOVL R0,W^MAC$GL_LINEPT ;SET UP LINE POINTER
0000'CF 50 D0 014F 268 MOVL R0,W^MAC$GL_ERRPTX ;RESET ERROR POINTER TO LINE START
OC 60 91 0154 269 CMPB (R0),#FF ;NEW SOURCE PAGE?
04 12 0157 270 BNEQ 60$ ;IF NEQ NO
0000'CF D6 0159 271 INCL W^MAC$GL_SRC_PAG ;YES--COUNT NEW PAGE
50 0000'CF C0 015D 272 ADDL2 W^MAC$GL_LINELN,R0 ;FIGURE ADDRESS OF LAST CHARACTER
60 OD 90 0162 273 MOVB #CR,(R0) ;STORE CR FOR END OF LINE
0000'CF D6 0165 274 INCL W^MAC$GL_LINENUM ;COUNT THIS LINE
04 6B OE E0 0169 275 BBS #FLG$V_P2,(R11),80$ ;BRANCH IF THIS IS PASS 2
0000'CF D6 016D 276 INCL W^MAC$GL_SRC_LCNT ;NO--COUNT LINE READ IN PASS 1
```



```
0000'CF 13 6B 51 D4 0171 279 80$:
0000'CF 0000'CF 27 E0 0171 280 CLRL R1 ; Initialise insert number
0000'CF 0000'CF 50 C3 0173 281 BBS #FLG$V UPDFIL,(R11),90$ ; Branch if file is being updated
0000'CF 0000'CF 10 E1 0177 282 SUBL3 W^MAC$GL_LINBAS, - ; Compute line number
0000'CF 0000'CF 19 E1 017E 283 W^MAC$GL_LINENUM,R0
0000'CF 0000'CF 10 D0 017F 284 BBC #FLG$V SEQFIL,(R11),100$ ; Branch if not sequenced file
0000'CF 0000'CF 10 11 0183 285 MOVL W^MAC$GL_RECHDBUF,R0 ; Get sequenced line number
0000'CF 0000'CF 10 11 0188 286 BRB 100$
0000'CF 0000'CF 10 3C 018A 287 90$:
0000'CF 0000'CF 02 E1 018A 288 MOVZWL W^MAC$GT_SCB+SUM_W_LINE_NO,R0 ; Get original line number
0000'CF 0000'CF 05 E1 018F 289 BBC #SUM_V_SRCUPD, - ; Branch if line from source file
0000'CF 0000'CF 05 3C 0194 290 W^MAC$GT_SCB+SUM_B_FLAGS,100$
0000'CF 0000'CF 05 3C 0195 291 MOVZWL W^MAC$GT_SCB+SUM_W_INSERT_NO,R1 ; Get insert number
0000'CF 0000'CF 50 B0 019A 292 100$:
0000'CF 0000'CF 51 B0 019A 293 MOVW R0,W^MAC$GW_LST_LINE ; Save listing line number
0000'CF 0000'CF 51 B0 019F 294 MOVW R1,W^MAC$GW_LST_INST ; and insert number
0000'CF 0000'CF 05 05 01A4 295 RSB
0000'CF 0000'CF 05 05 01A5 296
0000'CF 0000'CF 05 05 01A5 297 .DSABL LSB
0000'CF 0000'CF 05 05 01A5 298 :
0000'CF 0000'CF 05 05 01A5 299 :
0000'CF 0000'CF 05 05 01A5 300 SUM_ERROR:
0000'CF 0000'CF 04 D5 01A5 301 TSTL W^MAC$GL_LINENUM ; Error on very first line?
0000'CF 0000'CF 29 E2 01A9 302 BNEQ 5$ ; No if NEQ
0000'CF 0000'CF 29 E2 01AB 303 BBSS #FLG$V_FIRSTLN,(R11),5$
0000'CF 0000'CF 29 E2 01AF 304 5$:
0000'CF 0000'CF 22 D1 01AF 305 CMPL R0,#SUM$_EDITSCLSH ; Edits clash error?
0000'CF 0000'CF 22 12 01B6 306 BNEQ 4$ ; No if NEQ
0000'CF 0000'CF 03 E0 01B8 307 BBS #SUM_V_SUBCLSH, - ; Branch if not first clash
0000'CF 0000'CF 2F 2F 01BF 308 MAC$GT_SCB+SUM_B_FLAGS,10$
0000'CF 0000'CF 50 DD 01C0 309 PUSHL R0
0000'CF 0000'CF 50 DD 01C2 310 $INTOUT_X INT$_FNEWL ; Force new line
0000'CF 0000'CF 50 DD 01C8 311 POPL R0
0000'CF 0000'CF 50 DD 01CB 312 BISL2 #ST$$_INFO,R0 ; Make and information code
0000'CF 0000'CF 50 DD 01CE 313 $INTOUT_LW INT$_INFO,<R0,#0> ; Information message
0000'CF 0000'CF 50 DD 01DA 314 BRB 10$
0000'CF 0000'CF 50 DD 01DA 315 4$:
0000'CF 0000'CF 50 DD 01DA 316 PUSHL R0
0000'CF 0000'CF 50 DD 01DC 317 $INTOUT_X INT$_FNEWL ; Force new line
0000'CF 0000'CF 50 DD 01E2 318 POPL R0
0000'CF 0000'CF 50 DD 01E5 319 $INTOUT_LW INT$_WRN,<R0,#0> ; Warning message
0000'CF 0000'CF 50 DD 01EF 320 10$:
0000'CF 0000'CF 50 DD 01EF 321 $INTOUT_X INT$_SUME ; Source update merge error
0000'CF 0000'CF 50 DD 01F5 322 RSB
```



```
01F6 324 .SBTTL OPEN NEXT INPUT SOURCE FILE
01F6 325
01F6 326 :++
01F6 327 THIS ROUTINE IS CALLED TO OPEN THE NEXT INPUT FILE.
01F6 328 IF THE FILE IS SUCCESSFULLY OPENED AND CONNECTED,
01F6 329 RO IS SET TO 1. IF ANOTHER INPUT FILE CANNOT BE OPENED,
01F6 330 RO IS 0.
01F6 331
01F6 332 :--
01F6 333
01F6 334 MAC$NXTINPFIL:
01F6 335 PUSHF #^M<R1,R2,R3,R4,R5> ;SAVE REGISTERS USED BY MOVCS
01F8 336 PUSHAB W^MAC$GT_SCB ; Supply SUM control block address
01FC 337 CALLS #1,G^SUM$CLOSE ; Close any update files
0203 338 $DISCONNECT RAB=W^MAC$INPUT_RAB ;DISCONNECT THE RECORD ACCESS
020E 339 MOVL W^MAC$GL_CURINFDB,RO ;POINT TO CURRENT FDB
0213 340 $CLOSE FAB=8(RO) ;CLOSE THE INPUT FILE
021D 341 MOVL W^MAC$GL_LINENUM,- ;SET LINE BASE TO HIGHEST LINE NUMBER
0221 342 W^MAC$GL_LINBAS ;
0224 343 MOVCS #0,(SP),#^A/ /,#39,- ;BLANK FILL THE RESULTANT FILENAME
0229 344 W^MAC$AB_SBT_FILE ;
022C 345 MOVL W^MAC$GL_CURINFDB,RO ;GET ADDRESS OF FDB WE JUST CLOSED
0231 346 CMPL RO,W^MAC$GL_INPQUE+4 ;WAS IT THE LAST FDB
0236 347 BEQL 20$ ;IF EQL YES
0238 348 MOVL (RO),RO ;NO--LINK TO NEXT FDB
023B 349 BSBW MAC$OPEN_INPUT ;OPEN NEXT INPUT FILE
023E 350 POPR #^M<R1,R2,R3,R4,R5> ;RESTORE REGISTERS
0240 351 RSB ;RETURN WITH LBS IN RO
0241 352 :
0241 353 : NO MORE INPUT FILES
0241 354 :
0241 355 20$: CLRL RO ;INDICATE NO MORE FILES
0243 356 POPR #^M<R1,R2,R3,R4,R5>
0245 357 RSB
```



```
0246 359 .SBTTL STAT4 SWITCH INPUT TO MACRO TEXT
0246 360
0246 361 :++
0246 362 : FUNCTIONAL DESCRIPTION:
0246 363 :
0246 364 : THIS SEMANTIC ROUTINE SWITCHES THE POINTER TO THE CURRENT
0246 365 : INPU BLOCK TO POINT TO THE MACRO BEING EXPANDED. FIRST THE
0246 366 : REAL MACRO ARGUMENTS ARE SCANNED AND AN INPUT BLOCK IS CREATED.
0246 367 : THEN MAC$GETCHR WILL READ CHARACTERS FROM THE MACRO TEXT
0246 368 : EXPANDER MAC$GET_MAC_LIN.
0246 369 :
0246 370 :--
0246 371
00000000 372 .PSECT MAC$RO_CODE_P1,NOWRT,GBL, LONG
OE 0005'CF E8 0000 373 STAT4:: ; STATEMENT = MACTXT
56 0000'CF47 D0 0013 374 BLBS W^LST$G_MACROCALL+SYM$ VAL,10$ ;BRANCH IF LISTING MACRO CALLS
55 00'8F 9A 0019 375 $INTOUT_LW INT$ SETLONG,<#0,#MAC$GL LIST IT> ;NO--SEND FLAG TO PASS 2
FFEO' 30 001D 376 10$: MOVL W^MAC$AL VALSTACK[R7],R6 ;GET PTR TO MACRO MNB
05 A6 D5 0020 377 MOVZBL #CRFSK REF,R5 ;THIS IS A REFERENCE
1D 13 0023 378 BSBW MAC$CREF_MACRO ;CROSS REFERENCE IF CREFFING
0000'CF 56 D0 0025 379 TSTL MNB$ _TXTP(R6) ;IS THERE ANY TEXT
0000'CF 18 A6 D0 002A 380 BEQL 20$ ;IF EQL NO--TAKE THE QUICK OUT
FFCD' 30 0030 381 MOVL R6,W^MAC$GL MACPTR ;SAVE POINTER TO MACRO MNB
0000'CF 0000'CF D4 0033 382 MOVL MNB$ _ARGP(R6),W^MAC$GL_KEYMAC ;SET PTR TO KEYWORD ARGS (IF ANY)
00 6B 10 E3 0037 383 BSBW MAC$GET_ARGS ;SCAN REAL MACRO ARGS
5A OD 9A 0042 384 ;AND SET UP INPUT BLOCK
05 0045 385 CLRL W^MAC$GL_KEYMAC ;CLEAR POINTER TO KEYWORD ARGS
386 MOVL W^MAC$GL_BLKPTR,W^MAC$GL_INPUTP ;POINT TO NEW INPUT BLOCK
387 BBCS #FLGSV MACTXT,(R11),20$ ;FLAG READING MACRO TEXT
388 20$: MOVZBL #CR,R10 ;FORCE READING OF NEW LINE
389 30$: RSB
```



```
0046 391          .SBTTL MEXIT          MACRO EXIT ROUTINE
0046 392
0046 393 :++
0046 394 : FUNCTIONAL DESCRIPTION:
0046 395 :
0046 396 : THIS ROUTINE POPS THE INPUT CONTEXT ONE LEVEL TO EFFECT AN
0046 397 : EXIT FROM A MACRO OR REPEAT-TYPE MACRO
0046 398 :
0046 399 :--
0046 400
0046 401 MEXIT::                          ;DIRECTIVE = KMEXIT
0046 402
0046 403          .SBTTL MAC$POP_INPUT      POP INPUT CONTEXT UP A LEVEL
0046 404
0046 405 :++
0046 406 : FUNCTIONAL DESCRIPTION:
0046 407 :
0046 408 : THIS ROUTINE POPS THE INPUT CONTEXT A LEVEL. ALL PAGES
0046 409 : ALLOCATED TO THE CURRENT INPUT BLOCK ARE DEALLOCATED.
0046 410 :
0046 411 :--
0046 412
0046 413 MAC$POP_INPUT::
0046 414          PUSHL R6                      ;SAVE R6
0048 415          MOVL W^MAC$GL_INPUTP,R6    ;GET POINTER TO CURRENT INPUT BLOCK
004D 416          BBC #FLG$V_MACTXT,(R11),10$ ;BRANCH IF NOT READING MACRO TEXT
0051 417          MOVL INP$RPTCNT(R6),R0     ;YES--GET REPEAT COUNT (OR MACRO FLAG)
0055 418          BGEQ 10$                    ;IF GEQ NO NEED TO GO ANY FARTHER
0057 419          INCL R0                     ;SEE IF IT WAS REPEAT THAT WENT TO 0
0059 420          BEQL 10$                    ;IF EQL YES
005B 421          ;NO--IT WAS MACRO REDEFINING ITSELF
005B 422          BICL3 #^X80000000,INP$RPTCNT(R6),R6 ;YES--GET ADDRESS OF MNB
0064 423          BSBW MAC$DEL_MAC_DEF        ;AND DELETE THE MNB AND ASSOCIATES
0067 424          MOVL W^MAC$GL_INPUTP,R6    ;RE-GET THE INPUT BLOCK ADDRESS
006C 425 10$:          MOVL INP$LINK(R6),R0   ;POP INPUT CONTEXT
006F 426          MOVL R0,W^MAC$GL_INPUTP
0074 427          CMPL R0,#MAC$GL_P$MINBL    ;READING FROM SOURCE FILE?
007B 428          BNEQ 20$                    ;IF NEQ NO
007D 429          BBCC #FLG$V_MACTXT,(R11),20$ ;YES--CLEAR MACTXT FLAG
0081 430 20$:          MOVL INP$IFVAL(R6),W^MAC$GL_IF_VALUE ;POP IF VALUE
0087 431          MOVL INP$IFLVL(R6),W^MAC$GL_IF_LEVEL ;POP IF LEVEL
008D 432          MOVL R6,R0                  ;COPY INPUT BLOCK ADDRESS
0090 433          MOVL INP$PAGP(R6),R6        ;LINK TO ANY OTHER PAGES
0094 434          BSBW MAC$DEA_1_PAGE          ; Deallocate block (always 1 page)
0097 435          BRB 40$                     ;GO DEALLOCATE INPUT BLOCK AND REST
0099 436          ;OF INPUT CONTEXT PAGES
0099 437 30$:          MOVL MXB$LINK(R6),R6    ; Link to possible next page
009C 439          BSBW MAC$DEAL_BLOCK          ; Deallocate memory block
009F 440 40$:          MOVL R6,R0              ;POINT R0 FOR NEXT DEALLOCATION
009F 441          BNEQ 30$                     ;IF NEQ GO DO IT
00A2 442          MOVZBL #CR,R10              ;FORCE NEW LINE
00A4 443          POPL R6                      ;RESTORE R6
00A7 444          RSB
00AA 445
```

56	14	A6	80000000	8F	CB	005B	422	BICL3	#^X80000000,INP\$RPTCNT(R6),R6	;YES--GET ADDRESS OF MNB	
			FF99		30	0064	423	BSBW	MAC\$DEL_MAC_DEF	;AND DELETE THE MNB AND ASSOCIATES	
56			0000	CF	D0	0067	424	MOVL	W^MAC\$GL_INPUTP,R6	;RE-GET THE INPUT BLOCK ADDRESS	
			50	66	D0	006C	425	10\$: MOVL	INP\$LINK(R6),R0	;POP INPUT CONTEXT	
			0000	CF	50	D0	006F	426	MOVL	R0,W^MAC\$GL_INPUTP	
00000000			8F	50	D1	0074	427	CMPL	R0,#MAC\$GL_P\$MINBL	;READING FROM SOURCE FILE?	
			04	12	007B	428	BNEQ	20\$		;IF NEQ NO	
00	6B	10		E5	007D	429	BBCC	#FLG\$V_MACTXT,(R11),20\$	;YES--CLEAR MACTXT FLAG		
0000	CF	10	A6	D0	0081	430	20\$: MOVL	INP\$IFVAL(R6),W^MAC\$GL_IF_VALUE	;POP IF VALUE		
0000	CF	0C	A6	D0	0087	431	MOVL	INP\$IFLVL(R6),W^MAC\$GL_IF_LEVEL	;POP IF LEVEL		
		50	56	D0	008D	432	MOVL	R6,R0	;COPY INPUT BLOCK ADDRESS		
56	18	A6		D0	0090	433	MOVL	INP\$PAGP(R6),R6	;LINK TO ANY OTHER PAGES		
		FF69		30	0094	434	BSBW	MAC\$DEA_1_PAGE	; Deallocate block (always 1 page)		
		06	11	0097	435	BRB	40\$		;GO DEALLOCATE INPUT BLOCK AND REST		
				0099	436				;OF INPUT CONTEXT PAGES		
				0099	437	30\$: MOVL	MXB\$LINK(R6),R6		; Link to possible next page		
56	66			D0	0099	438					
	FF61			30	009C	439	BSBW	MAC\$DEAL_BLOCK	; Deallocate memory block		
					009F	440	40\$: MOVL	R6,R0	;POINT R0 FOR NEXT DEALLOCATION		
50	56			D0	009F	441					
	F5	12		00A2	442	BNEQ	30\$		;IF NEQ GO DO IT		
5A	0D	9A		00A4	443	MOVZBL	#CR,R10		;FORCE NEW LINE		
	56	8ED0		00A7	444	POPL	R6		;RESTORE R6		
		05		00AA	445	RSB					



```
00AB 447 .SBTTL STATEMENTS END-OF-LINE CLEANUP
00AB 448
00AB 449 :++
00AB 450 :
00AB 451 :--
00AB 452
00AB 453 :MTXT1::
00AB 454 :MTXT2::
00AB 455 MTXT3::
00AB 456 MTXT4::
00AB 457 MTEXT:
00AB 458
00AB 459 $INTOUT_X INT$ CHKL
00B1 460 BISL2 #FLG$M_BOL!FLG$M_EVALEXP!FLG$M_COMPEXP,- ;RESET BEGINNING OF LINE
00B7 461 (R11) ;AND EVALUATE FLAGS
00B8 462 CLRL W^MAC$GL_ABSFLAG ;RESET ABSOLUTE FLAG
00BC 463 CLRL W^MAC$GL_PRMSEG ;NO EXPRESSION PSECT YET
00C0 464 CLRL W^MAC$GL_MOPPTR ;CLEAR POINTER TO OPERAND MODE BYTES
00C4 465 BICL2 #FLG$M_OPRND!FLG$M_IFSTAT!FLG$M_NOREF,(R11) ;NOT IN OPERAND FIELD
00CB 466 ;AND NOT IN AN IF
00CB 467 ;AND ALLOW PRMSYM TO REF SYMBOLS
00CB 468 MOVB #RDX$V_DECIMAL,W^MAC$GB_RDXNDX ;RESET TO DECIMAL RADIX
00D0 469 RSB
00D1 470
00D1 471 .END
```

00000046 8F C8  
6B  
0000'CF D4  
0000'CF D4  
0000'CF D4  
6B 01802000 8F CA  
0000'CF 02 90  
05

00000046 8F C8  
6B  
0000'CF D4  
0000'CF D4  
0000'CF D4  
6B 01802000 8F CA  
0000'CF 02 90  
05

00AB 447 .SBTTL STATEMENTS END-OF-LINE CLEANUP  
00AB 448  
00AB 449 :++  
00AB 450 :  
00AB 451 :--  
00AB 452  
00AB 453 :MTXT1::  
00AB 454 :MTXT2::  
00AB 455 MTXT3::  
00AB 456 MTXT4::  
00AB 457 MTEXT:  
00AB 458  
00AB 459 \$INTOUT\_X INT\$ CHKL  
00B1 460 BISL2 #FLG\$M\_BOL!FLG\$M\_EVALEXP!FLG\$M\_COMPEXP,- ;RESET BEGINNING OF LINE  
00B7 461 (R11) ;AND EVALUATE FLAGS  
00B8 462 CLRL W^MAC\$GL\_ABSFLAG ;RESET ABSOLUTE FLAG  
00BC 463 CLRL W^MAC\$GL\_PRMSEG ;NO EXPRESSION PSECT YET  
00C0 464 CLRL W^MAC\$GL\_MOPPTR ;CLEAR POINTER TO OPERAND MODE BYTES  
00C4 465 BICL2 #FLG\$M\_OPRND!FLG\$M\_IFSTAT!FLG\$M\_NOREF,(R11) ;NOT IN OPERAND FIELD  
00CB 466 ;AND NOT IN AN IF  
00CB 467 ;AND ALLOW PRMSYM TO REF SYMBOLS  
00CB 468 MOVB #RDX\$V\_DECIMAL,W^MAC\$GB\_RDXNDX ;RESET TO DECIMAL RADIX  
00D0 469 RSB  
00D1 470  
00D1 471 .END



MAC\$INPUT  
Symbol table

GET NEXT CHARACTER

B 1

16-SEP-1984 02:20:18 VAX/VMS Macro V04-00  
5-SEP-1984 01:48:32 [MACRO.SRC]INPUT.MAR;1

Page 13  
(8)

\$\$TMP1 = 00000001  
 \$\$TMP2 = 000000A0  
 \$COUNT = 0000003B  
 ARG\$K\_SIZE = 000003E8  
 AUD\$K\_SIZE = 00000010  
 BIT... = 00000005  
 BLNK = 00000020  
 CHR\$M\_COMMA CR = 00000020  
 CHR\$M\_ILL CHR = 00000040  
 CHR\$M\_NUM BER = 00000010  
 CHR\$M\_SPA\_MSK = 00000001  
 CHR\$M\_SYM\_CH1 = 00000008  
 CHR\$M\_SYM\_CHR = 00000004  
 CHR\$M\_SYM\_DLM = 00000002  
 CHR\$V\_COMMA CR = 00000005  
 CHR\$V\_CVTLC = 00000061  
 CHR\$V\_ILL CHR = 00000006  
 CHR\$V\_NOCVT = 0000007F  
 CHR\$V\_NUM BER = 00000004  
 CHR\$V\_SPA\_MSK = 00000000  
 CHR\$V\_SYM\_CH1 = 00000003  
 CHR\$V\_SYM\_CHR = 00000002  
 CHR\$V\_SYM\_DLM = 00000001  
 CNT = 00000002  
 CR = 0000000D  
 CRFSK\_REF = \*\*\*\*\*  
 ERR = 00000000  
 FF = 0000000C  
 FLG\$M\_ALLCHR = 00000001  
 FLG\$M\_BOL = 00000002  
 FLG\$M\_CHKLPND = 00100000  
 FLG\$M\_COMPEXPR = 00000004  
 FLG\$M\_CONT = 00000008  
 FLG\$M\_CRF = 40000000  
 FLG\$M\_CRSEEN = 00000001  
 FLG\$M\_DATRPT = 00000010  
 FLG\$M\_DBGOUT = 00004000  
 FLG\$M\_DLIMSTR = 00008000  
 FLG\$M\_ENDMCH = 00000020  
 FLG\$M\_EVALEXPR = 00000040  
 FLG\$M\_EXPOPT = 00000080  
 FLG\$M\_EXTERR = 00010000  
 FLG\$M\_EXTWRN = 00020000  
 FLG\$M\_FIRSTLN = 00000200  
 FLG\$M\_IFSTAT = 00800000  
 FLG\$M\_IIF = 00400000  
 FLG\$M\_INSERT = 00000100  
 FLG\$M\_IRPC = 20000000  
 FLG\$M\_LEXOP = 00000002  
 FLG\$M\_LSTXST = 00000200  
 FLG\$M\_MAC2COL = 00000800  
 FLG\$M\_MACL = 00000800  
 FLG\$M\_MACLTB = 08000000  
 FLG\$M\_MACTXT = 00010000  
 FLG\$M\_MEBLST = 00001000  
 FLG\$M\_MOREARG = 00002000  
 FLG\$M\_MOREINP = 00000008

X 04

FLG\$M\_NEWPND = 00000400  
 FLG\$M\_NOREF = 01000000  
 FLG\$M\_NTTYPEPC = 00000020  
 FLG\$M\_NULCHR = 00040000  
 FLG\$M\_OBJXST = 00200000  
 FLG\$M\_OPNDCHK = 00000100  
 FLG\$M\_OPRND = 00002000  
 FLG\$M\_OPTVFLIDX = 00001000  
 FLG\$M\_ORDLST = 00020000  
 FLG\$M\_P2 = 00004000  
 FLG\$M\_RPTIRP = 10000000  
 FLG\$M\_SEQFIL = 02000000  
 FLG\$M\_SKAN = 00008000  
 FLG\$M\_SPECOP = 00000004  
 FLG\$M\_SPLALL = 04000000  
 FLG\$M\_STOIMF = 00040000  
 FLG\$M\_SYM2COL = 00000400  
 FLG\$M\_TOCFLG = 00080000  
 FLG\$M\_UPAFILG = 00000010  
 FLG\$M\_UPDFIL = 00000080  
 FLG\$M\_UPMARG = 00000040  
 FLG\$M\_XCRF = 80000000  
 FLG\$V\_ALLCHR = 00000000  
 FLG\$V\_BOL = 00000001  
 FLG\$V\_CHKLPND = 00000014  
 FLG\$V\_COMPEXPR = 00000002  
 FLG\$V\_CONT = 00000003  
 FLG\$V\_CRF = 0000001E  
 FLG\$V\_CRSEEN = 00000020  
 FLG\$V\_DATRPT = 00000004  
 FLG\$V\_DBGOUT = 0000002E  
 FLG\$V\_DLIMSTR = 0000002F  
 FLG\$V\_ENDMCH = 00000005  
 FLG\$V\_EVALEXPR = 00000006  
 FLG\$V\_EXPOPT = 00000007  
 FLG\$V\_EXTERR = 00000030  
 FLG\$V\_EXTWRN = 00000031  
 FLG\$V\_FIRSTLN = 00000029  
 FLG\$V\_IFSTAT = 00000017  
 FLG\$V\_IIF = 00000016  
 FLG\$V\_INSERT = 00000008  
 FLG\$V\_IRPC = 0000001D  
 FLG\$V\_LEXOP = 00000021  
 FLG\$V\_LSTXST = 00000009  
 FLG\$V\_MAC2COL = 0000002B  
 FLG\$V\_MACL = 0000000B  
 FLG\$V\_MACLTB = 0000001B  
 FLG\$V\_MACTXT = 00000010  
 FLG\$V\_MEBLST = 0000000C  
 FLG\$V\_MOREARG = 0000002D  
 FLG\$V\_MOREINP = 00000023  
 FLG\$V\_NEWPND = 0000000A  
 FLG\$V\_NOREF = 00000018  
 FLG\$V\_NTTYPEPC = 00000025  
 FLG\$V\_NULCHR = 00000032  
 FLG\$V\_OBJXST = 00000015  
 FLG\$V\_OPNDCHK = 00000028



MAC\$INPUT  
Symbol table

GET NEXT CHARACTER

C 1

16-SEP-1984 02:20:18 VAX/VMS Macro V04-00  
5-SEP-1984 01:48:32 [MACRO.SRC]INPUT.MAR;1

Page 14  
(8)

FLGSV\_OPRND = 0000000D  
FLGSV\_OPTVFLIDX = 0000002C  
FLGSV\_ORDLST = 00000011  
FLGSV\_P2 = 0000000E  
FLGSV\_RPTIRP = 0000001C  
FLGSV\_SEQFIL = 00000019  
FLGSV\_SKAN = 0000000F  
FLGSV\_SPECOP = 00000022  
FLGSV\_SPLALL = 0000001A  
FLGSV\_STOIMF = 00000012  
FLGSV\_SYM2COL = 0000002A  
FLGSV\_TOCLFG = 00000013  
FLGSV\_UPAFLG = 00000024  
FLGSV\_UPDFIL = 00000027  
FLGSV\_UPMARG = 00000026  
FLGSV\_XCRF = 0000001F  
HASHSZ = 0000007F  
HYPHEN = 0000002D  
INPSB\_ARGCT = 0000001C  
INPSK\_BLKSI2 = 00000021  
INPSK\_BUFSIZ = 000003E8  
INPSK\_IRPSIZ = 0000003C  
INPSL\_ARGS = 0000001D  
INPSL\_GETL = 00000008  
INPSL\_IFLVL = 0000000C  
INPSL\_IFVAL = 00000010  
INPSL\_LINK = 00000000  
INPSL\_NXTL = 00000004  
INPSL\_PAGP = 00000018  
INPSL\_RPTCNT = 00000014  
INTSK\_BUFSIZ = 000013F4  
INTSK\_BUFWRN = 00001390  
INTS\_ADD = 00000001  
INTS\_AND = 00000002  
INTS\_ASH = 00000003  
INTS\_ASN = 0000000C  
INTS\_AUGPC = 0000000D  
INTS\_BDST = 0000000E  
INTS\_CHKL = 0000000F  
INTS\_DIV = 00000004  
INTS\_END = 00000010  
INTS\_EPT = 00000011  
INTS\_ERR = 00000012  
INTS\_ETX = 00000013  
INTS\_FNEWL = 00000014  
INTS\_ILG = 00000000  
INTS\_INFO = 0000003A  
INTS\_LGLAB = 00000015  
INTS\_MACL = 00000016  
INTS\_MUL = 00000005  
INTS\_NEG = 00000006  
INTS\_NEWL = 00000017  
INTS\_NEWP = 00000018  
INTS\_NOT = 00000007  
INTS\_OP = 00000019  
INTS\_OR = 00000008  
INTS\_PRIL = 0000001A

INTS\_PRT = 0000001B  
INTS\_PSECT = 0000001C  
INTS\_REDEF = 0000001D  
INTS\_REF = 0000001E  
INTS\_REST = 0000001F  
INTS\_SAME = 00000009  
INTS\_SAVE = 00000020  
INTS\_SBTTL = 00000021  
INTS\_SETFLAG = 00000022  
INTS\_SETLONG = 00000023  
INTS\_SPIC = 00000024  
INTS\_SPID = 00000025  
INTS\_STIB = 00000026  
INTS\_STIL = 00000028  
INTS\_STIW = 00000027  
INTS\_STKEPT = 00000029  
INTS\_STKG = 0000002A  
INTS\_STKL = 0000002B  
INTS\_STKPC = 0000002C  
INTS\_STKS = 0000002D  
INTS\_STCB = 00000034  
INTS\_STOL = 0000002E  
INTS\_STOW = 00000035  
INTS\_STRB = 0000002F  
INTS\_STRL = 00000031  
INTS\_STRSB = 00000032  
INTS\_STRSW = 00000033  
INTS\_STRW = 00000030  
INTS\_STSB = 00000036  
INTS\_STSW = 00000037  
INTS\_SUB = 0000000A  
INTS\_SUME = 00000039  
INTS\_WRN = 00000038  
INTS\_XOR = 0000000B  
LSTSG\_MACROCALL = \*\*\*\*\* X 04  
LSTSK\_BUFSIZ = 00000086  
LSTSK\_L\_P\_PAGE = 0000003C  
LSTSK\_TITLE\_SIZ = 00000028  
MABSB\_ARGNO = 00000005  
MABSB\_NAME = 00000004  
MABSK\_BLKSI2 = 0000000C  
MABSL\_DVPTR = 00000008  
MABSL\_LINK = 00000000  
MABSW\_DVLEN = 00000006  
MAC\$AB\_LINEBF \*\*\*\*\* X 03  
MAC\$AB\_SBT\_FILE \*\*\*\*\* X 03  
MAC\$AL\_VALSTACK \*\*\*\*\* X 04  
MAC\$CREF\_MACRO \*\*\*\*\* X 04  
MAC\$DEAL\_BLOCK \*\*\*\*\* X 04  
MAC\$DEA\_T\_PAGE \*\*\*\*\* X 04  
MAC\$DEL\_MAC\_DEF \*\*\*\*\* X 04  
MAC\$GB\_RDXNDX \*\*\*\*\* X 04  
MAC\$GETCHR 00000000 RG 03  
MAC\$GETLIN 000000BB RG 03  
MAC\$GET\_ARGS \*\*\*\*\* X 04  
MAC\$GL\_ABSFLAG \*\*\*\*\* X 04  
MAC\$GL\_BLKPTR \*\*\*\*\* X 04

MA  
VO



MAC\$INPUT  
Symbol table

GET NEXT CHARACTER

D 1

16-SEP-1984 02:20:18 VAX/VMS Macro V04-00  
5-SEP-1984 01:48:32 [MACRO.SRC]INPUT.MAR;1

Page 15  
(8)

MAC\$GL_CURINFDB	*****	X	03
MAC\$GL_ERRPTX	*****	X	03
MAC\$GL_IF_LEVEL	*****	X	03
MAC\$GL_IF_VALUE	*****	X	04
MAC\$GL_INPQUE	*****	X	03
MAC\$GL_INPUTP	*****	X	03
MAC\$GL_KEYMAC	*****	X	04
MAC\$GL_LINBAS	*****	X	03
MAC\$GL_LINELN	*****	X	03
MAC\$GL_LINENUM	*****	X	03
MAC\$GL_LINEPT	*****	X	03
MAC\$GL_LIST_IT	*****	X	04
MAC\$GL_MACPTR	*****	X	04
MAC\$GL_MOPPTR	*****	X	04
MAC\$GL_PRMINBL	*****	X	04
MAC\$GL_PRMSG	*****	X	04
MAC\$GL_RECHDBUF	*****	X	03
MAC\$GL_SRCFAG	*****	X	03
MAC\$GL_SRC_LCNT	*****	X	03
MAC\$GT_SCB	*****	X	03
MAC\$GW_LST_INST	*****	X	03
MAC\$GW_LST_LINE	*****	X	03
MAC\$INPUT_RAB	*****	X	03
MAC\$INTERR_2_LW	*****	X	03
MAC\$INTOUT_2_LW	*****	X	03
MAC\$INTOUT_WD	*****	X	03
MAC\$INTOUT_X	*****	X	03
MAC\$NXTINPFIL	000001F6	R	03
MAC\$OPEN_INPUT	*****	X	03
MAC\$POP_INPUT	00000046	RG	04
MAC\$MISSINGEND	= 007D8828		
MAC\$SUBSYS	= 0000007D		
MEXIT	00000046	RG	04
MNBSB_ARGCT	00000017		
MNBSB_NAME	00000004		
MNBSK_BLKSI2	0000001C		
MNBSL_ARGP	00000018		
MNBSL_CRSYMF	00000013		
MNBSL_LINK	00000000		
MNBSL_PAGC	0000000F		
MNBSL_PAGP	0000000B		
MNBSL_TXTP	00000005		
MNBSW_FLAG	00000009		
MTEXT	000000AB	R	04
MTXT3	000000AB	RG	04
MTXT4	000000AB	RG	04
MXBSK_BLKSI2	00000008		
MXBSL_LINK	00000000		
MXBSL_PAGES	00000004		
OBJ\$K_BUF\$SI2	= 00000200		
OPF\$M_LASTOPR	= 00002000		
OPF\$M_OPTEXP	= 00001000		
OPF\$V_LASTOPR	= 0000000D		
OPF\$V_OPTEXP	= 0000000C		
PSC\$B_NAME	00000004		
PSC\$B_SEG	0000000C		
PSC\$B_UNUSED	0000000B		

PSC\$K_BLKSI2	00000013
PSC\$K_NO_OPTS	= 0000000A
PSC\$L_CURLOC	0000000F
PSC\$L_LINK	00000000
PSC\$L_MAXLGTH	00000005
PSC\$M_ABS	= FFFFFFFF7
PSC\$M_ALIGNFLG	= 00004000
PSC\$M_ALLOPTNS	= 000003FF
PSC\$M_BYTE	= 00004000
PSC\$M_CON	= FFFFFFFFB
PSC\$M_DEFAULT	= 000001C8
PSC\$M_EXE	= 000000C0
PSC\$M_GBL	= 00000010
PSC\$M_LCL	= FFFFFFFEF
PSC\$M_LIB	= 00000002
PSC\$M_LONG	= 00004800
PSC\$M_NOEXE	= FFFFFFFBF
PSC\$M_NOPIC	= FFFFFFFFE
PSC\$M_NORD	= FFFFFFFF7F
PSC\$M_NOSHR	= FFFFFFFDF
PSC\$M_NOVEC	= FFFFFFFDF
PSC\$M_NOWRT	= FFFFFFFEF
PSC\$M_OVR	= 00000004
PSC\$M_PAGE	= 00006400
PSC\$M_PIC	= 00000001
PSC\$M_QUAD	= 00004C00
PSC\$M_RD	= 00000080
PSC\$M_REL	= 00000008
PSC\$M_SHR	= 00000020
PSC\$M_USR	= FFFFFFFFD
PSC\$M_VEC	= 00000200
PSC\$M_WORD	= 00004400
PSC\$M_WRT	= 00000180
PSC\$S_ALIGNMENT	= 00000004
PSC\$V_ALIGNFLG	= 0000000E
PSC\$V_ALIGNMENT	= 0000000A
PSC\$V_EXE	= 00000006
PSC\$V_GBL	= 00000004
PSC\$V_LIB	= 00000001
PSC\$V_OVR	= 00000002
PSC\$V_PIC	= 00000000
PSC\$V_RD	= 00000007
PSC\$V_REL	= 00000003
PSC\$V_SHR	= 00000005
PSC\$V_VEC	= 00000009
PSC\$V_WRT	= 00000008
PSC\$W_FLAG	00000009
PSC\$W_OPTIONS	0000000D
RAB\$W_RSZ	= 00000022
RDX\$V_BINARY	= 00000000
RDX\$V_DECIMAL	= 00000002
RDX\$V_DOUBLE	= 00000005
RDX\$V_FLOAT	= 00000004
RDX\$V_GFLOAT	= 00000006
RDX\$V_HEX	= 00000003
RDX\$V_HFLOAT	= 00000007
RDX\$V_OCTAL	= 00000001



REGS_PC	=	0000000F		
RMS\$ EOF		*****	X	03
SEMI	=	0000003B		
SIZ...	=	00000001		
STAT4		00000000	RG	04
STBSK_PG MISS	=	0000000A		
STSSK_INFO	=	00000003		
STSSH_SEVERITY	=	00000007		
STSSS_FAC_NO	=	0000000C		
STSSV_FAC_NO	=	00000010		
SUMSCLOSE		*****	X	03
SUMSLINE		*****	X	03
SUMS_EDITSCLSH	=	00848800		
SUMS_NORMAL	=	00848001		
SUM_B_FLAGS		0000001C		
SUM_ERROR		000001A5	R	03
SUM_K_BLN		0000001D		
SUM_L_ISDATA		00000004		
SUM_L_STS		00000000		
SUM_M_AUDIT	=	00000001		
SUM_M_AUDITNEW	=	00000002		
SUM_M_DELETE	=	00000010		
SUM_M_SRCUPD	=	00000004		
SUM_M_SUBCLSH	=	00000008		
SUM_Q_AUDDS		00000008		
SUM_Q_FILESP		00000010		
SUM_V_AUDIT	=	00000000		
SUM_V_AUDITNEW	=	00000001		
SUM_V_DELETE	=	00000004		
SUM_V_SRCUPD	=	00000002		
SUM_V_SUBCLSH	=	00000003		
SUM_W_INSERT NO		0000001A		
SUM_W_LINE_NO		00000018		
SYMSB_NAME		00000004		
SYMSB_SEG		0000000C		
SYMSB_TOKEN		0000000B		
SYMSK_BLKSIZ		0000000D		
SYMSK_MAXLEN	=	0000001F		
SYMSK_TWOCOL	=	00000010		
SYMSL_LINK		00000000		
SYMSL_VAL		00000005		
SYMSM_ABS	=	00000010		
SYMSM_ASN	=	00000100		
SYMSM_CRFO	=	00002000		
SYMSM_DEBUG	=	00000020		
SYMSM_DEF	=	00000001		
SYMSM_DELMAC	=	00000200		
SYMSM_EPT	=	00000200		
SYMSM_EXTRN	=	00000008		
SYMSM_GLOBL	=	00000004		
SYMSM_LOCAL	=	00000040		
SYMSM_ODBG	=	00000000		
SYMSM_REF	=	00000080		
SYMSM_RELPSECT	=	00000800		
SYMSM_SUPR	=	00004000		
SYMSM_WEAK	=	00000002		
SYMSM_XCRF	=	00001000		

```
SYMSV_ABS = 00000004  
SYMSV_ASN = 00000008  
SYMSV_CRFO = 0000000D  
SYMSV_DEBUG = 00000005  
SYMSV_DEF = 00000000  
SYMSV_DELMAC = 00000009  
SYMSV_EPT = 00000009  
SYMSV_EXTRN = 00000003  
SYMSV_GLOBL = 00000002  
SYMSV_LOCAL = 00000006  
SYMSV_ODBG = 0000000A  
SYMSV_REF = 00000007  
SYMSV_RELPSECT = 0000000B  
SYMSV_SUPR = 0000000E  
SYMSV_WEAK = 00000001  
SYMSV_XCRF = 0000000C  
SYMSW_FLAG 00000009  
SYS$CLOSE ***** GX 03  
SYS$DISCONNECT ***** GX 03  
TAB = 00000009  
X1 = 00000400  
X2 = 0000000F
```



+-----+  
! Psect synopsis !  
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
. BLANK .	00000000 ( 0.)	01 ( 1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$AB\$\$	0000003C ( 60.)	02 ( 2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
MAC\$RO_CODE_P15	00000246 ( 582.)	03 ( 3.)	NOPIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC LONG
MAC\$RO_CODE_P1	000000D1 ( 209.)	04 ( 4.)	NOPIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC LONG

+-----+  
! Performance indicators !  
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.02	00:00:01.67
Command processing	107	00:00:00.48	00:00:04.12
Pass 1	263	00:00:04.89	00:00:24.79
Symbol table sort	0	00:00:00.65	00:00:02.10
Pass 2	102	00:00:01.14	00:00:04.59
Symbol table output	50	00:00:00.23	00:00:00.43
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	555	00:00:07.43	00:00:37.72

The working set limit was 1500 pages.  
42445 bytes (83 pages) of virtual memory were used to buffer the intermediate code.  
There were 40 pages of symbol table space allocated to hold 652 non-local and 38 local symbols.  
471 source lines were read in Pass 1, producing 22 object records in Pass 2.  
25 pages of virtual memory were used to define 23 macros.

+-----+  
! Macro library statistics !  
+-----+

Macro library name	Macros defined
-\$255\$DUA28:[SHRLIB]SUM.MLB;1	3
-\$255\$DUA28:[MACRO.OBJ]MACRO.MLB;1	12
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	9
TOTALS (all libraries)	24

827 GETS were required to define 24 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:INPUT/OBJ=OBJ\$:INPUT MSRC\$:INPUT/UPDATE=(ENH\$:INPUT)+LIB\$:MACRO/LIB+SHRLIB\$:SUM/LIB



0225 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY



0226 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

